UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR
INGENIUS
Revista de Ciencia y Tecnología

PlaM-DeP: una plataforma modular para el desarrollo y evaluación de algoritmos de detección de plagio académico

Hernán Fajardo Heras[1], Manuel Barrera Maura[2], Vladimir Robles Bykbaev[3], Cristian Timbi Sisalima[4,*] y Eduardo Calle Ortiz[5]

Resumen

En este trabajo se presenta un modelo de plataforma de *software* para desarrollar y evaluar los algoritmos de detección de plagio. La plataforma se basa en un diseño modular escalable, que implementa un conjunto de servicios que posibilitan realizar automáticamente tareas como: análisis sintáctico y semántico a través de WordNet y FreeLing, extracción automática de texto de múltiples formatos de archivos (PDF, Word y texto), extracción de contenido de páginas web (empleando algunos motores de búsqueda como Google, Yandex, Yahoo, Bing), el almacenamiento, la carga y el uso de algoritmos de detección de plagio. Estos servicios permiten a un programador desarrollar el código centrando el esfuerzo en el diseño del algoritmo y la base matemática/estadística. Actualmente, la plataforma se probó usando varias consultas de texto (n-gramas), y los resultados de rendimiento son prometedores.

Palabras clave: detección, FreeLing, MultiWordNet, plagio, plataforma plagio.

Abstract

In this paper we present a software platform model to develop and evaluate plagiarism detection algorithms. The platform is based in a scalable modular design, and implements several services to perform automatically the following tasks: syntactic and semantic analysis through WordNet and FreeLing, automatic text extraction of multiple file formats (PDF, Word and text), web page content extraction (using some search engines like Google, Yandex, Yahoo, Bing), and storage, load and use of plagiarism detection algorithms. These services allow a programmer to develop a code focusing the effort on the design of the algorithm and the mathematical/statistical basis. The platform was tested using several text queries (n-grams), and currently the performance results are promising.

Keywords: detection, FreeLing, MultiWordNet, plagiarism, plagiarism platform.

[1] Colaborador del Grupo de Investigación en Sistemas Informáticos e Inteligencia Artificial, Carrera de Ingeniería de Sistemas, Universidad Politécnica Salesiana, sede Cuenca.
[2] Colaborador del Grupo de Investigación en Sistemas Informáticos e Inteligencia Artificial, Carrera de Ingeniería de Sistemas, Universidad Politécnica Salesiana, sede Cuenca.
[3] Máster en Inteligencia Artificial Reconocimiento de Formas e Imagen Digital, Ingeniero en Sistemas, estudiante de Doctorado en Tecnologías de la Información y las Comunicaciones – Universidad de Vigo, Coordinador del Grupo de Investigación en Sistemas Informáticos e Inteligencia Artificial, UPS, sede Cuenca.
[4,*] Máster en Project Management, Ingeniero en Sistemas, Docente Investigador del Grupo de Investigación en Sistemas Informáticos e Inteligencia Artificial, UPS, sede Cuenca. Autor para correspondencia ✉: ctimbi@ups.edu.ec
[5] Máster en Tecnologías de la Información en Fabricación, Director del Centro de Investigación, Desarrollo e Innovación en Ingeniería, UPS, sede Cuenca.

Recibido: 25-04-2014, Aprobado tras revisión: 14-05-2014.

Forma sugerida de citación: Fajardo, H., Barrera, M., Robles, V., Timbi, C. y Calle, E. (2014). "PlaM-DeP: una plataforma modular para el desarrollo y evaluación de algoritmos de detección de plagio académico". Ingenius. N.°11, (Enero-Junio). pp. 32-41. ISSN: 1390-650X.

1. Introducción

Hoy en día la mayoría de autores definen de forma general el plagio como una copia de ideas, pensamientos u obras y su correspondiente presentación o publicación como propias. Sin embargo, consideramos como más adecuada la definición que plantea el IEEE (Instituto de Ingenieros Eléctricos y Electrónicos, por sus siglas en inglés), que indica lo siguiente: *"plagiar es reusar las ideas, procesos, resultados o palabras de alguien más, sin mencionar explícitamente la fuente y su autor"* [1]. Este concepto representa para nuestra propuesta un aspecto fundamental, ya que las técnicas de detección que se implementarán a través de la plataforma están estrechamente relacionadas con el análisis de referencias bibliográficas, el análisis de textos (sinonimia, textual, frecuencia de aparición, etc.) y el análisis de patrones regulares.

Es importante aclarar que no en todos los casos el hecho de tomar textos y parafrasearlos o reusarlos sin incluir una referencia se considera plagio, siempre y cuando sean de dominio general [2]. Un caso concreto de ello es el uso de fechas o acontecimientos públicos, por ejemplo, en un texto se puede tener la siguiente frase: *"La batalla de Pichincha ocurrió el 24 de mayo de 1822"* y dicha frase puede estar presente en muchos otros textos. En virtud de ello, en este tipo de casos no se puede referenciar a un autor que haya tenido la idea original, pero en el caso de que se tratase de una opinión o interpretación de estos hechos, como son las recitaciones, editoriales, artículos de análisis, etc. y en los que sí existen ideas propias de los autores, debería existir la referencia correspondiente.

A continuación mencionaremos dos casos de plagio, el primero se trata del periodista Fareed Zakaria, quien trabajaba en The New York Times y CNN y que admitió haber plagiado algunos párrafos de un ensayo de la profesora Lepore, para insertarlos en su artículo. Como consecuencia del plagio cometido al periodista le suspendieron su programa de televisión en CCN y su columna en el The New York Times fue suspendida por un mes [3]. El segundo caso se trata del plagio de un artículo de la Dra. Gabriela Piriz Álvarez, el cual fue publicado en la revista médica Uruguay el 20 de marzo del 2004, dos años después partes de este artículo, como la introducción, 16 párrafos y 2 bibliografías, aparecieron publicadas en un artículo en Internet, que pertenecía al Departamento de Salud del Gobierno de Navarra. La Dra. Gabriela Piriz les hizo saber que uno de sus artículos contenía plagio, pero lo único que hizo la institución fue cambiar algunas frases, pero el plagio aún continuaba, cabe destacar que el artículo jamás fue retirado [4].

Otro dato interesante señala que, según una encuesta realizada por la ATL (Association of Teachers and Lecturers) en el 2008 a profesores de escuelas de Gran Bretaña, el 58% de los profesores consideran el plagio como un problema serio, y el 28% de estos docentes indicaron que al menos el 50% de los trabajos entregados contenían plagio de Internet, incluso afirmaron que algunos trabajos llegaban con anuncios de las páginas web [5]. De esto se puede deducir que en este tipo de casos los estudiantes no se tomaron el tiempo para leer el contenido del trabajo que presentaron.

De igual forma, podemos decir que estos casos son ejemplos claros del síndrome de *"copy-paste"* planteado por Hermann Maurer y Narayanan Kulathuramaiyer, autores que indican que el acceso al amplio contenido de información en la Web es un factor que degrada la calidad de los trabajos científicos [6]. De acuerdo a esta misma encuesta la ATL indicó que más del 50% de los profesores afirmaron que los estudiantes no tienen comprensión de lo que es el plagio [5].

En virtud de lo expuesto, es fundamental contar con una herramienta que permita realizar el análisis automático en la detección de plagio, aspecto que permitirá mejorar los procesos educativos, y a la vez, creará una cultura de respeto hacia los trabajos e ideas de los demás.

El resto del presente artículo se organiza de la siguiente forma: en la sección 2 se revisan los aspectos relacionados con los tipos de detección de plagio, las herramientas y las tareas fundamentales que se deben efectuar durante este proceso; en la sección 3 se presenta la arquitectura de la plataforma propuesta; los resultados de rendimiento en cuanto a consultas basadas en n-gramas y tiempos de respuesta se analizan en la sección 4; finalmente, en la sección 5 se revisan las conclusiones y trabajo futuro.

2. Breve revisión del estado del arte de las técnicas de detección de plagio

Al momento, se ha desarrollado una amplia gama de herramientas informáticas para detectar plagio en textos y en otro tipo de trabajos, sin embargo, es fundamental mencionar que no existe una herramienta que certifique que efectúa la detección con un 100% de precisión [7]. Esto se debe principalmente a la complejidad que poseen los textos, a la aplicación de técnicas como el parafraseo, sinonimia o el uso de traductores. Por ello, se debe tener presente que estos sistemas deben considerarse como tecnologías de soporte y al momento no tienen la palabra final al tomar acciones contra quien ha cometido plagio [8]. Esta tarea siempre será responsabilidad de quienes usan las herramientas, en el caso del área académica será labor de los docentes el asegurarse por estos y otros medios si existe o no plagio.

Es importante mencionar que el análisis de plagio puede realizarse de dos formas:

En línea. Se efectúa a través de páginas web que permiten cargar el documento que se desea verificar, luego estas buscan a través de Internet documentos que tengan coincidencia con el sospechoso y al cabo de un tiempo determinado, nos presentan los resultados del análisis en la misma página o a través de correo electrónico.

Local. No requiere contar con Internet para realizar la revisión, por lo que se tiene que instalar una herramienta informática en nuestro computador y los documentos que van a ser comparados con el sospechoso deben estar ubicados localmente.

Asimismo, los tipos de herramientas de detección pueden ser de libre distribución, de código abierto o privativo y nos proveen los siguientes servicios:

- Análisis de grandes cantidades de documentos de manera rápida.
- Análisis automático del estilo de escritura en base a técnicas de inteligencia artificial u otras similares, lo que potencia su efectividad [9].
- Identificación de los fragmentos plagiados tanto en el documento original como en el sospechoso [9].
- Amplio soporte a diferentes formatos de documentos de texto, esto incluye documentos en Word, PDF, etc.
- Ahorro de tiempo en la revisión de casos de plagio, posibilitando que docentes y revisores se concentren en evaluar la calidad del contenido de los trabajos [10].

2.1. Técnicas de detección de plagio

En la actualidad se han desarrollado diferentes tipos de técnicas para la detección de plagio en textos, unas con mejores resultados que otras, se puede decir que estas técnicas realizan su análisis desde diferentes enfoques, entre los que están [8]:

- Detección de plagio intrínseco.
- Detección de plagio con comparación a fuentes externas.

A continuación se detallan algunas de las técnicas más importantes de acuerdo a estos enfoques.

2.1.1. Detección de plagio intrínseco

Este tipo de detección se realiza utilizando características del estilo de escritura, obtenidas a partir del mismo documento. Para este análisis no es necesario disponer de fuentes externas con las cuales comparar el texto, sin embargo, conlleva una mayor complejidad en los algoritmos para detectar plagio, verbigracia, la aplicación de técnicas de inteligencia artificial (como el Procesamiento del Lenguaje Natural). Este método en lugar de identificar las posibles fuentes, busca determinar la probabilidad de que el documento contenga información plagiada [9], [11].

Técnica de detección por estilo del autor

Este tipo de técnica intenta detectar el plagio en un documento en función del estilo de escritura del autor, es decir, busca partes de texto dentro del documento que se sospecha que tienen plagio y que poseen otro estilo de escritura. El gran problema que presenta este método, es que es necesario contar a priori con el estilo de escritura del autor del documento. Esta técnica es muy utilizada cuando realizamos una detección intrínseca, es decir, no contamos con fuentes externas para comparar con el documento sospechoso de plagio, la búsqueda de partes sospechosas se realiza a partir del mismo documento [11].

El uso de redes neuronales artificiales puede ser una herramienta adecuada para poder reconocer el estilo de escritura de un determinado autor, esto se logra cuantificando determinados aspectos del estilo del autor, como por ejemplo, el número de signos de puntuación, palabras más usadas, errores gramaticales que suelen cometer, entre otros.

Técnica de detección de cambios de complejidad

Este tipo de técnica consiste en detectar cambios en la complejidad del texto dentro de un fragmento del documento, esto se logra a través de la comparación del estilo de escritura de dicho fragmento con el estilo de escritura del resto del documento, cuando existe un cambio brusco que sobrepasa un umbral tolerado, se considera que ese fragmento es una posible inserción de información de fuentes externas y de no estar citado, se considera un posible plagio [9].

2.1.2. Detección de plagio con comparación a fuentes externas

Este tipo de técnicas conllevan el tener disponibilidad de acceso a las posibles fuentes de donde pudo existir plagio, este enfoque permite identificar de manera clara las fuentes y fragmentos plagiados, si es que existiese, pero realizando un análisis comparativo entre el documento sospechoso y dichas fuentes [8].

Es importante aclarar que las fuentes externas pueden ser de diferentes tipos, esto incluye documentos publicados en la web e incluso puede ser un corpus de

trabajos de diferentes estudiantes para determinar si existe copia entre estos.

2.2. Tareas comunes en los procesos de detección de plagio

Al momento de efectuar un proceso de detección de plagio, existen diversas tareas que se deben realizar para determinar si se presentan fragmentos de texto no referenciados. Por ello, si se revisan las propuestas que plantean diversos autores para el desarrollo de técnicas de detección [12], [13], [14], podemos identificar las siguientes tareas básicas:

- Aplicación de técnicas de procesamiento de lenguaje natural. En esta tarea se realizan operaciones base como eliminación de *stopwords* (palabras que no poseen carga semántica, como son artículos, preposiciones, etc.), análisis morfológico, análisis sintáctico, etc.
- Consulta de bases de datos léxicas y tesauros. En determinadas actividades se requiere consultar sinónimos, lemas, significados y otras características de las palabras, a fin de realizar operaciones de análisis textual más complejo.
- Análisis de texto en varios idiomas. La detección de plagio multilingüe [12] es un ámbito donde se requieren realizar tareas de traducción entre idiomas.
- Consulta de bases de datos de fuentes de información. Para poder detectar de dónde se han tomado pasajes de texto, se debe contar las fuentes originales. Estas fuentes pueden ser archivos locales o documentos almacenados en Internet.
- Extracción de texto del archivo a analizar. En esta etapa se extrae información textual o imágenes, dependiendo qué tipo de análisis de plagio se efectúe.

3. PlaM-DeP: modelo y arquitectura base

La plataforma tiene como objetivo brindar de forma automatizada la mayor parte de las funcionalidades descritas en la sección 2.2. Con ello, los desarrolladores de nuevos algoritmos de detección de plagio pueden centrar su esfuerzo en el diseño, funcionalidad y en el análisis matemático o estadístico que subyace en la técnica que se desea implementar. Asimismo, también se brinda la posibilidad para que se puedan evaluar algoritmos ya desarrollados, a fin de establecer el nivel de precisión o cobertura que poseen.

[1] http://nlp.lsi.upc.edu/freeling/

[2] http://multiwordnet.fbk.eu/english/home.php

La plataforma posee un esquema basado en interfaces genéricas que permiten realizar la "inyección" de los algoritmos y con ello realizar su carga de forma dinámica, todo ello en forma de librerías externas. Otro aspecto importante es que el diseño del sistema está pensado para ser implementado en un lenguaje de programación orientado objetos, por lo cual su primera versión se ha desarrollado en Java, ya que soporta la mayor parte de las bases de la programación orientada a objetos definidas por [15]. Este lenguaje permite agregar los algoritmos en tiempo de ejecución usando componentes conocidos como cargadores de clase (*Class Loader*), para lo cual se requiere una versión compilada del mismo.

3.1. Diseño

Como se aprecia en la Figura 1, la plataforma está diseñada bajo un enfoque modular y posee cuatro módulos principales: administración y gestión automatizada de servicios, análisis semántico, extracción de información textual y conexión a Internet y *web crawling*. Al trabajar bajo este enfoque, es posible agregar más módulos que nos permitan incorporar nuevas funcionalidades, como la de traducción automática (para análisis de plagio multilingüe). Del mismo modo, es factible realizar cambios en los módulos sin que ello afecte al funcionamiento general, verbigracia, es factible reemplazar los buscadores usados, las herramientas para análisis léxico, etc.

A continuación se detalla de forma breve las funcionalidades y características de los módulos.

Módulo de análisis léxico y semántico

Como se puede apreciar en la Figura 2, este módulo se basa en FreeLing[1] y una versión que hemos modificado de MultiWordNet[2]. Los servicios de procesamiento de texto que provee este módulo son extensibles y se realizan a través de la interfaz ***IAnalisisUtils***:

- Diccionario de sinónimos
- Diccionario de significados
- Diccionario de español-inglés
- Detección del tipo de palabra (sustantivo, adjetivo, verbo, etc.)
- Detección de párrafos
- Eliminación de stopwords
- Cálculo de la palabra con mayor frecuencia de aparición

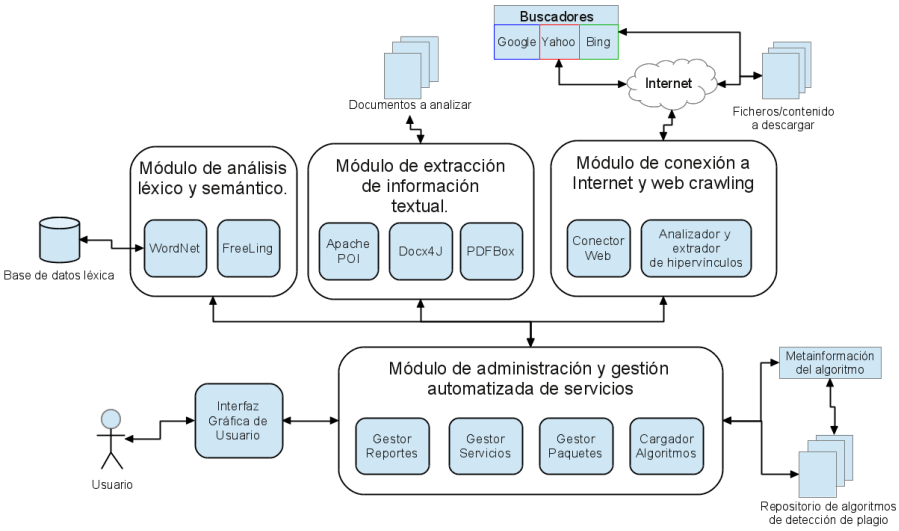

Figura 1. Vista modular general de la plataforma PlaM-DeP v1.0.

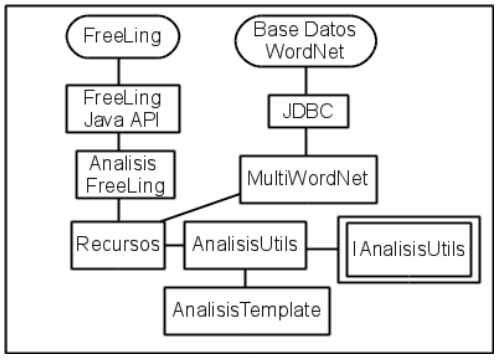

Figura 2. Componentes del módulo de análisis léxico y semántico.

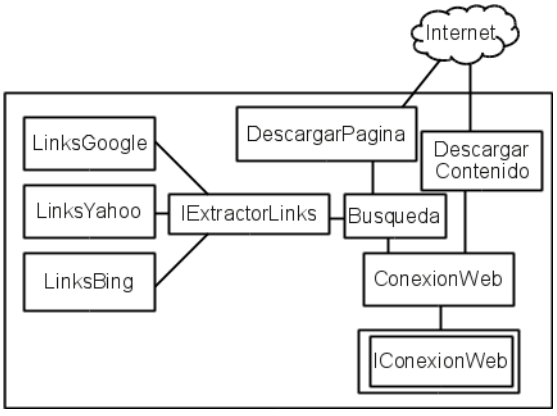

Figura 3. Componentes del módulo conexión a Internet y *web crawling*.

Módulo de análisis léxico y semántico

Este módulo se encarga de proveer los servicios para realizar las búsquedas en Internet y la descarga de contenido de las páginas, algunas de sus funcionalidades son:

- Realizar búsquedas en los principales motores (Google, Yahoo, Bing).
- Análisis y descarga del código fuente de la página de resultados que devuelve el motor.
- Descargar el texto de una página web o documento en Internet que representa la posible fuente de donde se han tomado pasajes.
- Extracción automatizada de los enlaces y su correspondiente descarga para análisis local.

Como se señala en la Figura 3, a través de la interfaz *IConexionWeb* se realizan las siguientes operaciones para recuperar las fuentes de búsqueda de plagio externo:

- Se construye la URL para realizar la consulta al motor de búsqueda seleccionado y se descarga la página que devuelve dicho motor.
- Se analiza la página extraída a fin de obtener las direcciones de los documentos que son la posible fuente de donde se han tomado los pasajes de texto. Esta tarea se realiza a través de expresiones regulares.
- Las direcciones son almacenadas en el *buffer* de memoria temporal y se proceda a la descarga de cada uno de los documentos, descargas que son distribuidas de manera paralela a fin de optimizar el uso del ancho de banda de la conexión a Internet.
- A partir de los documentos descargados, se procede a su análisis. Si son páginas web se usa un Parser HTML para extraer su contenido, y si son documentos en formato DOC, DOCX, PDF o TXT, se reutiliza la funcionalidad provista por el módulo de extracción de información textual.

Módulo de extracción de información textual

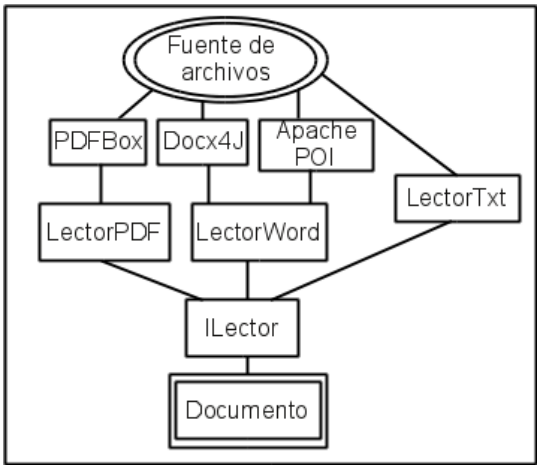

Figura 4. Componentes del módulo de extracción de información textual.

Este módulo provee la funcionalidad para realizar la lectura de documentos en diferentes formatos (PDF, DOC, DOCX o TXT). Para efectuar este proceso, se trabaja con la interfaz ***ILector***, que recibe la ruta del documento, detecta el formato de archivo y emplea las librerías PDF Box, Docx4J o Apache POI para devolver el contenido en forma de cadenas de texto. Si el documento se encontrase en otro formato, el módulo lo leería como texto plano.

Módulo de administración y gestión automatizada de servicios

Este módulo se encarga de integrar todos los módulos de servicios anteriormente mencionados, además, es el que permite cargar los algoritmos y ejecutarlos de acuerdo a las peticiones del usuario.

Como se observa en la Figura 5, los módulos de servicios se integran a la plataforma a través de interfaces, y un submódulo se encarga de brindar acceso a los servicios ofrecidos. Además, la plataforma inyecta los objetos necesarios a los algoritmos de dos formas:

Como objetos compartidos: Son servicios que se instancian una sola vez a nivel de toda la plataforma, y se comparten declarándose con alcance clasificador (static), dichos servicios son:

- Servicio de acceso a motores de búsqueda de Internet
- Servicio de análisis léxico.

Como múltiples instancias: En algunos casos se requiere realizar procesos paralelos como la descarga de documentos de Internet, en este caso para cada ejecución de un algoritmo se realiza una nueva instancia. El mismo caso sucede con la lectura de archivos desde disco, básicamente se tienen múltiples instancias de los siguientes servicios:

- Servicio de descarga de documentos de Internet.
- Servicio de lectura de documentos en disco.

Es importante mencionar que los procesos paralelos a los que se hace referencia se presentan cuando existen múltiples ejecuciones del algoritmo de detección o se generan varias instancias de diferentes algoritmos en paralelo.

Carga de algoritmos

Todos los algoritmos que se pretenden cargar a la plataforma deben heredar de una clase abstracta, la cual obliga que dicho algoritmo implemente los métodos de comparación y de retorno de resultados de acuerdo a este. Asimismo, los algoritmos deben estar registrados en una base de datos para que la plataforma pueda cargarlos, en dicho registro se deberá especificar principalmente la ruta del ejecutable del algoritmo, el nombre del paquete y el nombre de la clase que implementa los métodos heredados de la clase abstracta.

Cuando arranca la plataforma, automáticamente accede a la base de datos y de acuerdo a la información recuperada carga los algoritmos como librerías extras usando cargadores de clases. Todas las instancias de los algoritmos cargados se pasan a un contenedor, el mismo que permite que más adelante se lo invoque y se obtenga una nueva instancia sin necesidad de volver a cargarlo.

Ejecución de algoritmos

Para que un usuario pueda acceder y ejecutar un algoritmo de detección desde cualquier interfaz gráfica, se debe enviar básicamente los siguientes parámetros:

- Nombre del algoritmo a utilizar
- Tipo de comparación (búsqueda en Internet, contra archivos locales o los dos casos)
- La ruta del documento a ser analizado
- El listado de las rutas de los documentos considerados como posibles fuentes.

Una vez que se reciben estos parámetros, la plataforma accede al contenedor de algoritmos y obtiene una nueva instancia a la que se le inyectan los servicios, se le pasa el contenido del documento a ser analizado y dependiendo de lo especificado por el usuario, le pasa las posibles fuentes que pueden ser obtenidas localmente o desde Internet. La plataforma invoca al método de comparación del algoritmo y obtiene los resultados para generar un informe, que se almacena como archivo PDF.

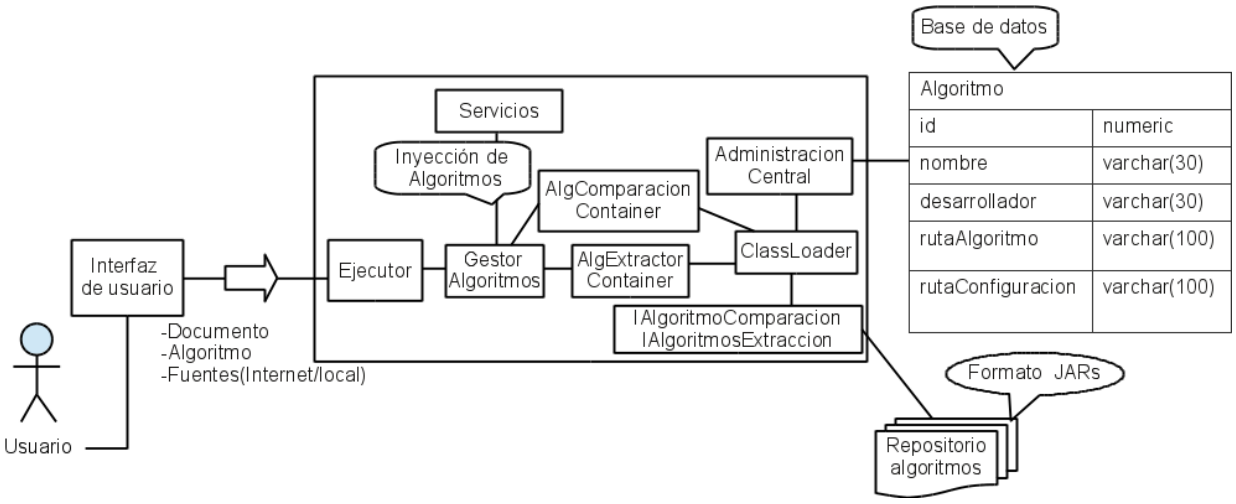

Figura 5. Componentes del módulo de administración y gestión automatizada de servicios

4. Análisis de resultados

A fin de verificar el correcto funcionamiento de la plataforma, se simuló un proceso de análisis de plagio real y para ello se enviaron múltiples consultas usando n-gramas de 1 a 10 palabras. En la Figura 6 se aprecian los tiempos de respuesta obtenidos por los tres motores de búsqueda empleados. Se puede observar que Bing posee tiempos de respuesta del orden de los 800 ms, Google de 2000 ms y Yahoo de 2500 ms.

Una segunda batería de pruebas que se realizó consistió en usar n-gramas para analizar documentos completos. De esta forma, la plataforma requiere una hora aproximada para procesar 1000 n-gramas. Sin embargo, se debe recalcar que el proceso dependerá del ancho de banda disponible; además, esta prueba se hizo de manera que las peticiones se procesan en cola, es decir, si se dispone de acceso a Internet de alta velocidad, el trabajo se podría realizar de manera paralela.

Adicionalmente, se realizaron pruebas para medir la velocidad de carga y extracción de información textual de los archivos que se desea analizar. En la Tabla 1 se presenta una cuadro comparativo de la velocidad de lectura de acuerdo al formato con un texto de 5272 palabras y 35254 caracteres.

Tabla 1. Comparativa en la velocidad de carga de un archivo de 5272 palabras, de acuerdo a cuatro tipos de formato: PDF, DOC, DOCX y texto plano

Formato	**Velocidad de carga a la plataforma**
PDF	948 ms
DOC	514 ms
DOCX	4371 ms
Texto plano	81 ms

4.1. Pruebas en el analizador léxico semántico

Al igual que en los casos anteriores, también se evaluaron algunas de las funcionalidades del módulo de análisis léxico utilizando textos y palabras de prueba. Dichos resultados se detallan a continuación.

Texto de pruebas

El texto usado para pruebas es el que se transcribe en el siguiente párrafo:

> *El plagio tiene una gran diversidad de clasificaciones, que pueden incluir diferentes áreas o tipos de obras, por ejemplo plagio en obras musicales, obras literarias, imágenes, etc. pero en este trabajo se procurará centrarse en los principales tipos de plagio en textos y se detallan a continuación*

Análisis de un documento

El analizador es capaz de reconocer y separar el documento en párrafos y llevar a cabo diversas operaciones. A continuación presentamos algunas de las más utilizadas y el resultado que producen cada una de ellas:

Eliminación de *Stop Words*
plagio tiene gran diversidad clasificaciones, pueden incluir diferentes áreas tipos obras por ejemplo plagio obras musicales obras literarias imágenes pero este trabajo se procurará centrarse principales tipos de plagio en textos y se detallan a continuación

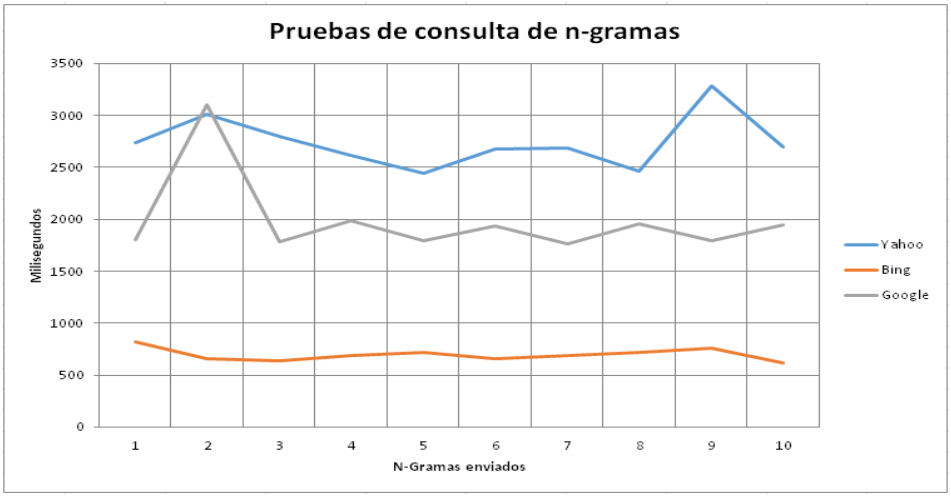

Figura 6. Tiempo de respuesta de los buscadores empleados en la plataforma usando n-gramas (de 1 a 10).

Tokenizar en NGramas por ejemplo (bi-gramas)
El plagio plagio tiene tiene una una gran...

Cálculo de la palabra más relevante de acuerdo a la mayor frecuencia de aparición:
argmax(fa) = plagio, Donde fa = frecuencia de aparición

Características Generales Palabra: mayoría Tipo: NCFS00 Lema: mayoría
Traducciones Bulk Majority Absolute_majority Legal_age Majority
Sinónimos Mayoría Mayoría absoluta
Antónimos Minoría
——Significados—— Más de la mitad de los votos

Adicionalmente se tienen funciones de preparación para el análisis que son llamadas de manera automática, por ejemplo, el eliminar caracteres duplicados como espacios, saltos de carro, etc. Otra funcionalidad de este tipo es eliminar los caracteres que no son alfanuméricos (por ejemplo: la coma, el punto y coma, etc.).

Análisis de un párrafo (palabra por palabra)

El análisis efectuado se basa inicialmente en reconocer los párrafos del documento, para luego empezar a analizar palabra por palabra. Por cada palabra que se encuentre en el texto, el analizador léxico es capaz de devolver sus características léxicas, como son la etiqueta Eagle (la cual es un código de representación morfológica de una palabra de acuerdo a FreeLing), el tiempo en el que se encuentra, si está en plural o singular, etc., además, se obtiene el lema, un listado de los sinónimos, antónimos, traducciones y posibles significados de esa palabra. Por ejemplo con la palabra "mayoría":

4.2. Prueba de un algoritmo de detección de plagio

Finalmente, se probó la plataforma con un algoritmo de detección de plagio basado en la técnica Vector Space Model (Modelo de espacios de vector) y similitud de cosenos, el cual consiste en representar cada documento como un vector en el espacio y buscar la diferencia entre estos usando la similitud de cosenos, cuyo resultado está en el rango de 0 a 1. La expresión matemática en la que se basa esta técnica es la siguiente (ecuación 1):

$$\cos\theta = \frac{\mathbf{d_2} \cdot \mathbf{q}}{\| \mathbf{d_2} \| \| \mathbf{q} \|} \quad (1)$$

En este caso, el proceso consiste en guardar todas las palabras del documento en un *hashtable*, así como también el número de veces que se repite, esto se

hará con los dos documentos que queremos comparar. Luego de ello, en el *hash* se quedarán solo las palabras comunes entre los dos documentos, para finalmente aplicar similitud de coseno entre los dos vectores como se muestra en la Figura 7 y verificar el porcentaje de plagio.

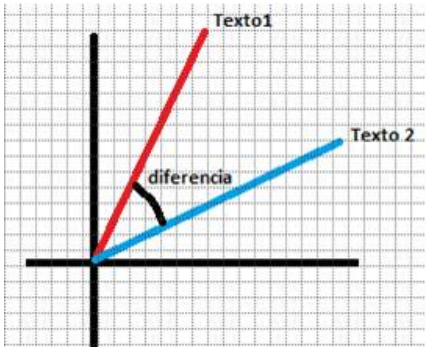

Figura 7. Vectores con palabras comunes a los dos textos.

A continuación se muestra una pequeña parte de los textos comparados.

Texto 1

Historia

El creador de Facebook es Mark Zuckerberg, estudiante de la Universidad de Harvard. La compañía tiene sus oficinas centrales en Palo Alto, California.

La idea de crear una comunidad basada en la Web en que la gente compartiera sus gustos y sentimientos no es nueva, pues David Bohnett, creador de Geocities, la había incubado a fines de los años 1980. Una de las estrategias de Zuckerberg ha sido abrir la plataforma Facebook a otros desarrolladores.

Entre los años 2007 y 2008 se puso en marcha Facebook en español traducido por voluntarios,7 extendiéndose a los países de Latinoamérica. Casi cualquier persona con conocimientos informáticos básicos puede tener acceso a todo este mundo de comunidades virtuales.

Texto 2

Historia

Fue creado por estudiantes universitarios, se dice que originalmente Facebook no fue planeado sino que resulto más de lo que realmente se esperaba, los principales fundadores de esta red fueron: Mark Zuckerberg junto a Eduardo Saverin, Chris Hughes y Dustin Moskovitz siendo el primero el CEO actual de la empresa que es hoy Facebook.

Modelo de Negocio

Su negocio está basado en la información de los usuarios, es decir cuando los usuarios comparten sus gustos y preferencias, la red analiza esta información para ofrecer publicidad personalizada, además Facebook vende las estadísticas de preferencias a empresas grandes a quienes les interesa y sirve esta información.

Texto 3

El plagio tiene una gran diversidad de clasificaciones, que pueden incluir diferentes áreas o tipos de obras, por ejemplo plagio en obras musicales, obras literarias, imágenes, etc. pero en este trabajo se procurará centrarse en los principales tipos de plagio en textos y se detallan a continuación

Al comparar los textos se obtuvieron los siguientes resultados:

Documento 1	Documento 2	% Similitud
Texto 1	Texto 1	100%
Texto 2	Texto 2	100%
Texto 3	Texto 3	100%
Texto 1	Texto 2	255155%
Texto 1	Texto 3	20053%
Texto 2	Texto 3	0.0844%

Para contrastar se han comparado los documentos contra sí mismos para verificar que se detecte un 100% de similitud.

5. Conclusiones

En este trabajo se ha presentado una plataforma que provee varios de los servicios requeridos durante la ejecución de procesos de detección de plagio textual. Asimismo, se ha podido observar que el sistema se comporta de forma adecuada, posibilitando realizar miles de consultas de búsqueda de fuentes en Internet. El esquema bajo el que se ha construido la plataforma permite que cualquier usuario pueda descargar su código y realizar los ajustes que creyere conveniente.

Asimismo, es importante mencionar que un problema que se percibió respecto a las búsquedas en Internet es que algunos motores limitan el número de consultas que se les realiza. De esta forma, cuando se envía una gran cantidad de consultas, luego de un tiempo bloquean al usuario por un periodo que varía de 10 a 15 minutos y para solucionar este problema se ha optado por alternar la conexión a esos motores a través de *proxys web*.

6. Trabajo futuro

Como líneas de trabajo futuro se proponen las que se detallan a continuación:

Crear un módulo para brindar funcionalidades de traducción, a fin de contar con soporte para el desarrollo de algoritmos de análisis de plagio multilingüe.

Implementar el *frontend* de la plataforma orientado como un servicio en línea, que pueda ser usado por la comunidad académica como una herramienta de detección de plagio de documentos.

Agradecimientos

Este trabajo ha sido financiado por el proyecto de investigación *"Desarrollo de un sistema para detección de plagio académico e implementación de una plataforma para investigación de técnicas de detección (SISDEP) CIDII-010113"* , de la Universidad Politécnica Salesiana, sede Cuenca.

Referencias

[1] IEEE. Plagiarism. [Online]. Available: http://www.ieee.org/publications_standards/publications/rights/plagiarism_FAQ.html

[2] H. A. Maurer, F. Kappe, and B. Zaka, "Plagiarism-a survey." *Journal of Universal Computer Science*, vol. 12, no. 8, pp. 1050–1084, 2006.

[3] El Espectador. (2012) Suspenden a periodista de time y cnn por un caso de plagio. [Online]. Available: http://www.elespectador.com/impreso/cultura/medios/

[4] A. Rodríguez, "Plagios y fraudes en la era de la globalización," *Revista médica de Uruguay*, no. 22, pp. 83–86, 2006.

[5] ATL (Association of Teachers and Lecturers). (2008, January, 18) School work plagued by plagiarism - atl survey. [Online]. Available: http://www.atl.org.uk/media-office/media-archive/School-work-plagued-by-plagiarism-ATL-survey.asp

[6] H. Maurer. (2007, October 15) Narayanan kulathuramaiyer, coping with the copy-paste-syndrome. World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education. [Online]. Available: http://www.editlib.org/p/26479

[7] S. Urbina, R. de Ozollo, J. Gallardo, C. Martí, A. Torres, and M. Torrens. (2010) Análisis de herramientas para la detección del ciberplagio.

[8] A. Cedeño, M. Vila, and P. Rosso, "Detección automática de plagio, de la copia exacta a la paráfrasis," pp. 76–96, 2010.

[9] D. Rodríguez-Torrejón and J. Martín-Ramos, "Leap: Una referencia para la evaluación de sistemas de detección de plagio con enfoque intrínseco," *Universidad de Huelva*, pp. 1–12, 2012.

[10] Turnitin. Detector de plagio online. [Online]. Available: http://turnitin.com/es

[11] D. Fúnez and M. Errecalde, "Detección de plagio intrínseco usando la segmentación de texto," in *CACIC – XVII Congreso argentino de Ciencias de la Computación*, 2011, pp. 91–100.

[12] M. Potthast, A. Barrón-Cedeño, B. Stein, and P. Rosso, "Cross-language plagiarism detection," *Language Resources and Evaluation*, vol. 45, no. 1, pp. 45–62, 2011.

[13] M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso, "An evaluation framework for plagiarism detection," in *Proceedings of the 23rd international conference on computational linguistics: Posters*. Association for Computational Linguistics, 2010, pp. 997–1005.

[14] S. M. Alzahrani, N. Salim, and A. Abraham, "Understanding plagiarism linguistic patterns, textual features, and detection methods," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 2, pp. 133–149, 2012.

[15] M. Ortiz and A. Plaza, *Programación orientada a objetos con Java y UML*, 1st ed. Editorial Universitaria Abya-Yala, 2014.